Matti Taina

Data Center Monitoring Using Nagios

Helsinki Metropolia University of Applied Sciences
Bachelor of Engineering
Computer Engineering
Thesis
12 May 2011

Metropolia

Abstract

Author Title Number of Pages Date	Matti Taina Data Center Monitoring using Nagios 32 pages 12 May 2011
Degree	Bachelor of Science
Degree Programme	Computer Engineering
Specialisation option	Data Networks
Instructors	Ville Rindell, Production Manager Erik Pätynen, Senior Lecturer

The goal of this final year project was to document and further develop an existing Nagios monitoring installation at Otaverkko Oy. This report begins by going through the basics in network monitoring and the Nagios software and also digs deeper into specific monitoring needs for different network components.

The client for this project was Otaverkko Oy. Their existing monitoring system was already a well-developed and functioning entity, but there was also work to be done to make it better and more suitable for their current conditions. Monitoring for some devices was not as reliable as could have been possible and the monitoring for the other Nagios process was not functioning as it should have.

The outcome of the project included numerous monitoring scripts created for different devices from scratch. The scripts were created using the PHP programming language and the new features these scripts provided were documented to their relevant information systems. In addition, a simple failover check was implemented that monitors the health of the monitoring system itself.

Keywords	Nagios, network monitoring, data center

Metropolia

Tiivistelmä

Tekijä Otsikko Sivumäärä Aika	Matti Taina Konesalin valvonta Nagios-ohjelmiston avulla 32 sivua 12.5.2011
Tutkinto	insinööri (AMK)
Koulutusohjelma	tietotekniikka
Suuntautumisvaihtoehto	tietoverkot
Ohjaajat	tuotantopäällikkö Ville Rindell lehtori Erik Pätynen

Tässä insinöörityössä tavoitteena oli dokumentoida ja kehittää Otaverkko Oy:n olemassaolevaa Nagios-ohjelmistolla toteutettua valvontajärjestelmää. Tässä raportissa käydään läpi verkonvalvonnan ja Nagios-ohjelmiston perusteita sekä perehdytään tarkemmin eri verkkolaitteiden yksityiskohtaisiin valvontatarpeisiin.

Opinnäytetyön toimeksiantaja oli Otaverkko Oy. Nykyisessä Nagios-asennuksessa paljon oli jo tehty tehokkaan valvonnan eteen, mutta myös kehittämisen varaa oli havaittavissa. Laitteiden valvonta ei toiminut kaikilta osin luotettavasti eikä toisen Nagios-prosessin valvonta toiminut halutulla tavalla.

Projektin tuloksena syntyi lukuisia valvontaskriptejä eri laitteille, joita ei vielä tarpeen mukaisesti valvottu. Jos valmista skriptiä ei löytynyt, työ tehtiin itse ottamalla selvää laitteesta ja kirjoittamalla Nagios-ohjelmiston kanssa yhteensopiva skripti PHP-ohjelmointikielellä. Skriptit laajensivat yrityksen teknisten asiantuntijoiden saamaa näkemystä laitteiden toimintatilasta ja sekä skriptit että itse laitteet dokumentoitiin asianmukaisiin järjestelmiin. Nagios-asennukseen tehtiin lisäksi yksinkertainen failover-tarkistus, joka valvoo itse hälytysjärjestelmän toimivuutta.

Avainsanat	Nagios, verkonvalvonta, konesali

Contents

Abbreviations

BGP	Border Gateway Protocol is a routing protocol used to connect portions of the Internet called autonomous systems to each other.
CPU	Central Processing Unit is the part of a computer that handles instructions sent by a computer program.
DNS	Domain Name System is a naming system that translates names into IP addresses.
FC	Fibre Channel is a fast network technology used for storage networking.
ICMP	Internet Control Message Protocol is a network protocol used primarily to indicate whether or not a network node is reachable.
IP	Internet Protocol is a telecommunications protocol that functions as the basis for networks such as the Internet.
ISP	Internet Service Provider is a company or an organization that provides access to the Internet or hosts services.
MSM	Management and Switching Module is a management module card used by Extreme Networks in their high-end switching routers.
OS	Operating System is software used as the basic user interface of a computer.
OSPF	Open Shortest Path First is a routing protocol used for interior routing and operates within a single autonomous system (AS).
SNMP	Simple Network Monitoring Protocol is a network management protocol widely used by active networking hardware.
SSH	Secure Shell is a secure remote management protocol that can be used instead of other insecure protocols such as Telnet or RSH (Remote Shell).
TCP	Transmission Control Protocol is connection-oriented protocol that many applications use to communicate over IP networks.
TLD	Top-level Domain is a domain name at the highest level in the Domain Name System.
UDP	User Datagram Protocol is a connectionless protocol used with applications that require fast response times and do not require error correction.

1 Introduction

Today's network services have become more and more performing and more critical at the same time. Constantly various different services are made outsourced and run by service providers with access to centralized server farms, or data centers. Outsourcing these services puts the responsibility of stability and technical awareness into the hands of service providers instead of the consumer or organization that is using the service. This obviously is costly for the client and hence great stability and usability are expected. Everything is running smoothly when one does not even notice the service provided underneath.

However, even though everything seems to run without problems and seems easy, the service provider may be working night and day and investing large amounts of money to make sure everything stays transparent. When the client sees just a simple Windows network share, what lies underneath is a storage appliance equipped with several terabytes of fast performing and high redundancy storage space, not to mention the gigabit-speed network that connects the client and the service.

When problems arise, it is imperative that the service provider is the one that sees this and reacts before the problem is noticable to the end-user. This means that the service provider has to be proactive and monitor on the equipment providing the service. Rather than walking around the server room and looking for a red light to come on, it is much more feasible to have it done automatically via a monitoring system. Because everything is connected to a network, it is easy to poll a computer system whether everything are going well.

The goal of this project is meant to document and improve the Nagios monitoring system used in Otaverkko Oy's data centers. Otaverkko Oy is a long time Internet service provider with three separate data centers and thousands of IPs and services monitored. The company also provides entire networking solutions with Internet connectivity for clients with the use of network switches and wireless access points and controllers, along with monitoring services. This document aims to cover the basics of network monitoring with Nagios, as well as its usage at Otaverkko and the work I did in this project to improve their monitoring system installation.

2 Network monitoring techniques

2.1 Simple Network Management Protocol

2.1.1 Definition

The Simple Network Management Protocol (SNMP) was designed to let a management system communicate with a network node equipped with an SNMP agent. What an SNMP agent does, is collect information about a target device and expose that information to a management system by listening to incoming queries on a network IP address. Another way to communicate is using SNMP traps. In this scenario a network node unprompted sends out a message via a network to either a configured trap manager IP address or a network broadcast address, exposing the information to any trap manager configured correctly. To provide base-level security, SNMP constitutes the use of a community string attached to every message sent over the network. If an agent does not recognize the community defined by the manager, it does not reply. [1]

SNMP operates on the UDP protocol. An SNMP agent listens to port 161 for SNMP get and SNMP set requests. A get request is used for reading information and a set request for writing (or setting) values. SNMP can also be used to send out so-called trap messages. In this case a trap listener is placed on a network, and SNMP agents send messages either directly to the listener IP address or to a broadcast address. In SNMP versions 1 and 2c this information is transmitted in clear text over the network. Figure 1 shows an example traffic capture screenshot using the Wireshark traffic analyzing tool. [1]

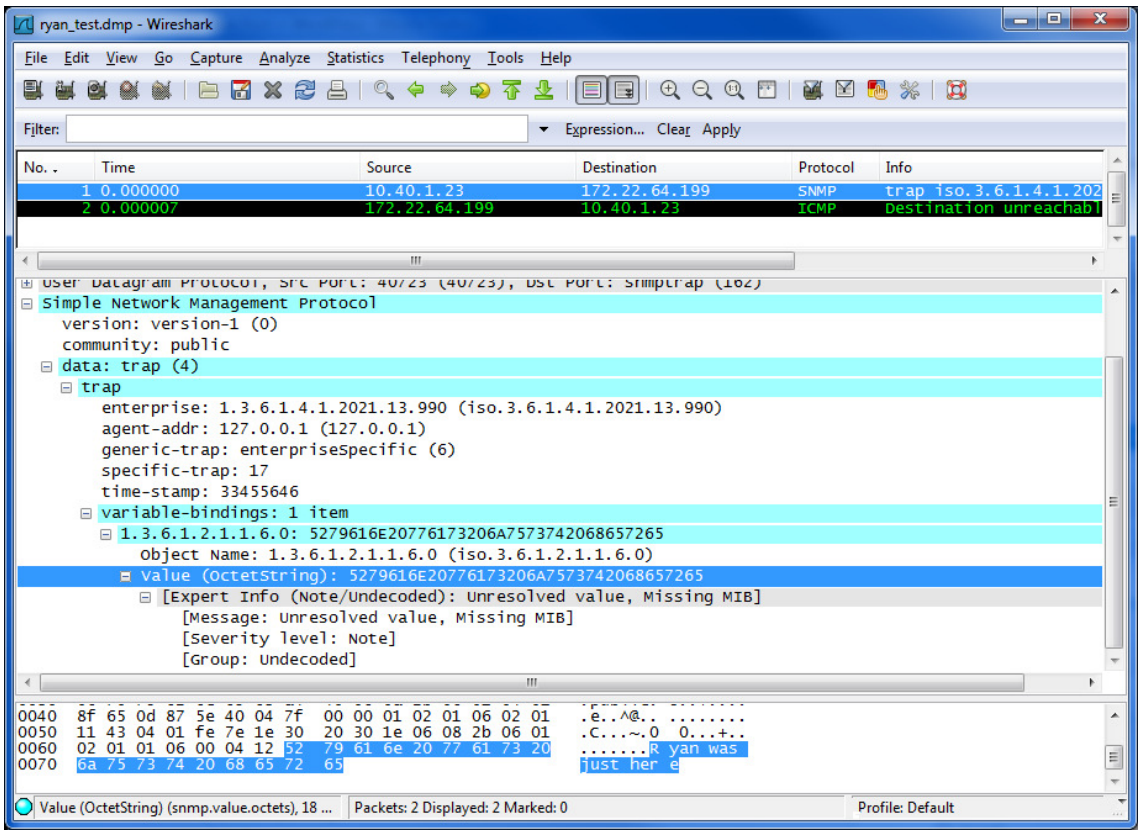

Figure 1: SNMP trap message captured and viewed by the Wireshark network analyzing tool. [2]

The content of the message is displayed hierarchically as layers in the center part of the figure and in hexadecimal and ASCII form on the bottom part. [1]

2.1.2 Management Information Base

SNMP works in conjunction with management information bases (MIBs), which are used for managing different types of equipment in a network. MIBs consist of objects defined using a subset of Abstract Syntax Notation One (ASN.1), called Structure of Management Information Version 2 (SMIv2). Objects in an MIB have an object identifier (OID), which can be presented in numeric form or as a string. For instance the numeric OID for the path to a device's uptime is .1.3.6.1.2.1.1.3 and the corresponding name for it is sysUpTime. [3]

Listing 1 provides a sample piece of an MIB file.

```
TOASTER-MIB DEFINITIONS ::= BEGIN
```

```
IMPORTS
        enterprises
                FROM RFC1155-SMI
        OBJECT-TYPE
                FROM RFC-1212
        DisplayString
                FROM RFC-1213;

epilogue        OBJECT IDENTIFIER ::= {enterprises 12}
toaster         OBJECT IDENTIFIER ::= {epilogue 2}


toasterManufacturer OBJECT-TYPE
  SYNTAX  DisplayString
  ACCESS  read-only
  STATUS  mandatory
  DESCRIPTION
          "The  name  of  the  toaster's  manufacturer.  For
instance,
          Microsoft Toaster."
  ::= {toaster 1}

toasterModelNumber OBJECT-TYPE
  SYNTAX  DisplayString
  ACCESS  read-only
  STATUS  mandatory
  DESCRIPTION
          "The name of the toaster's model. For instance,
          Radiant Automatic."
  ::= {toaster 2}
```

Listing 1. Sample piece of a MIB file.

MIB files are usually provided by hardware manufacturers to suit their own devices. An object identifier is always unique, and every manufacturer usually has their own private MIB (or several MIBs) in addition to the standardized MIBs defined by the Internet Engineering Task Force (IETF). MIBs are updated periodically to provide new functionality, remove obsolete entries and fix errors in previous versions. [3]

2.1.3 Versions

In a practical sense SNMP has three separate versions: version 1 (SNMPv1), version 2c (SNMPv2c, also known as version 1.5) and version 3 (SNMPv3). In version 2c changes and improvements are mostly cosmetic compared to version 1. The new version came with enhancements to packet handling and performance, but did not really have any new features. However version 3 introduced new features, such as packet encryption for data protection, message integrity to ensure the data is not modified along the way and authentication to verify the message source. [3]

In SNMPv3 security was much improved by getting rid of the community string and replacing it with a User-Based Security Model (USM). The new model in turn has three separate security levels: noAuthNoPriv, authNoPriv and authPriv. Between these levels one can choose to have authentication and privacy if one wants to. Table 1 shows the different levels and their features across all three SNMP versions. [4]

Table 1. SNMP security models and levels. [4]

Model	**Level**	**Authentication**	**Encryption**	**Description**
v1	noAuthNoPriv	Community string	None	Community string authentication
v2c	noAuthNoPriv	Community string	None	Community string authentication
v3	noAuthNoPriv	Username	None	Username authentication
v3	authNoPriv	MD5 or SHA	None	Hashed authentication
v3	authPriv	MD5 or SHA	DES	Hashed authentication and encryption

On the authPriv level two keys are needed: one for privacy (encrypting the message contents) and one for authentication. The password is hashed before sending it over the network using either HMAC-MD5-96 or HMAC-SHA-96 protocols, which make use of the MD5 and SHA1 hash functions. The hash output that is produced (along with a username) is then sent to the SNMP agent which matches it against a username-hash combination among its stored credentials. The content of the message is encrypted

using the DES encryption algorithm by taking the first eight octets of the privacy key that results after a successful authentication. [5, 4-5]

2.1.4 Usage in a Monitoring Scenario

SNMP is widely used in network monitoring because of its simplicity, open standard and performance. A vast number of network-connected devices have built-in SNMP support – not only for general remote management and monitoring benefit, but also for the manufacturers' own management software. SNMP can provide information about the device's software as well as hardware, including networking scheme and statistics. Any information that can be presented as a string or a number can be advertised via an SNMP agent. [1]

Basic SNMP operations are get, set and trap. These messages (excluding trap) handle only a single object identifier at a time. The get next operation is used to read single OIDs iteratively, sending consecutive requests. SNMP version 2c introduced a new operation called get bulk, which enables the transporting of multiple OIDs in one message. The snmpwalk and snmptable command line tools utilize the get next and get bulk operations, depending on the version of SNMP being used. [3]

A network management station (NMS) is often used to perform SNMP get and set requests and also to receive trap messages. Hardware manufacturers usually provide their own software for this purpose, although common usage software is also available. [6]

2.2 Strategies

Network monitoring cannot be trusted to a single piece of hardware in a mission-critical environment. Even though collecting information via SNMP and other means may not require much performance, redundancy plays an important role when it comes to trusting the network with automation. If one only has a single machine responsible for alerting the technicians of a problem, nothing will tell whether that single machine and its monitoring software is actually operational. For this reason it is imperative to have at least two independent machines dedicated for alerting staff of a problem. In addition to just being separated into different machines, it is important to place them

as far away from each other in a topological sense. This means that they must not share a network juncture or a power source. Having done this, one will still get notified even if one of the machines goes down altogether because of a power failure.

Another point to consider when planning one's network monitoring needs is the size of the network. If the network is spread over a geographically large area with a great number of hardware, it is probably sensible to delegate monitoring duties to several machines spread across the network in key places. The benefits with this strategy is that Wide Area Network (WAN) links over great distances will not be bloated with monitoring traffic, but instead they will only be used to monitoring the other endpoint's monitoring server. In addition this also provides much needed redundancy to the monitoring system. [Ville Rindell, Production Manager, 8 March 2011, personal communication].

3 Nagios

3.1 History

The history of Nagios started with Ethan Galstad, a Computer Science graduate out of Minnesota, USA – also known as the Father of Nagios. In 1996 Galstad created a simple MS-DOS program that used other third-party applications to ping remote network nodes and report the result as numeric pages. Pinging means sending an ICMP Echo Request packet over a network to a recipient, and seeing whether the other party replies with an ICMP Echo Reply packet. Two years later Galstad started building a more sophisticated application to run on the Linux operating system. This turned into an open-source project called NetSaint, that later was renamed to Nagios for legal reasons. The new name was actually an acronym for Nagios Ain't Gonna Insist On Sainthood. Later on Nagios would form to be a must-have tool in enterprise data networks. Figure 2 shows an example screen capture of the Nagios web interface Tactical Overview page. [7]

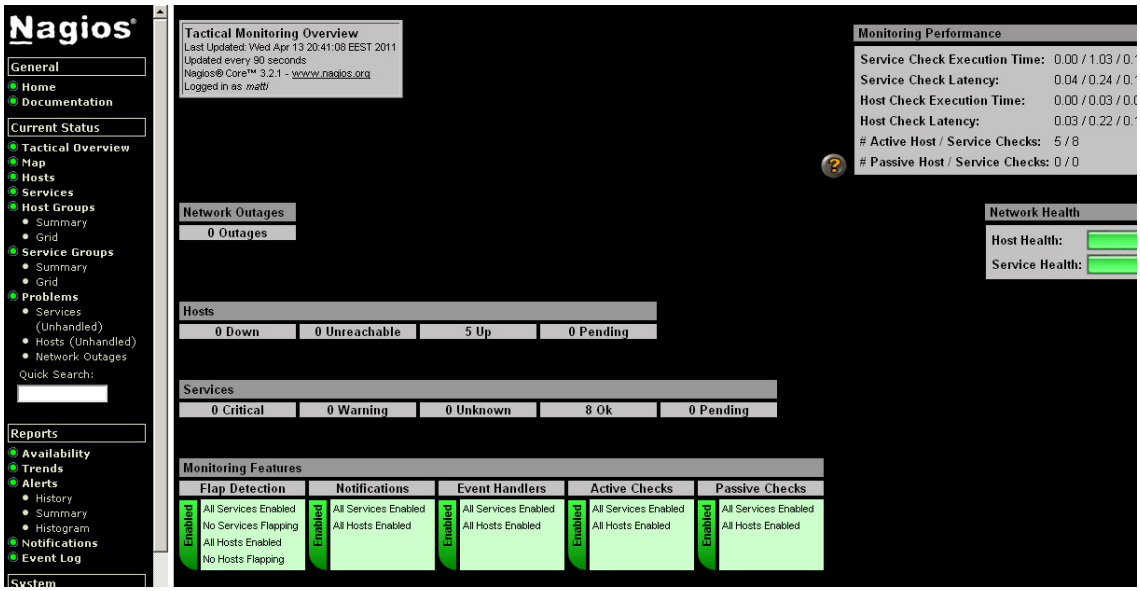

Figure 2: Nagios web interface Tactical Overview page.

Otaverkko has been using Nagios since it was still called NetSaint. Then the software was still very simple, and did not scale well with many monitored IP addresses. The checks were all done at the same time, which meant that the server running NetSaint

came under great load and produced bursts in network traffic whenever it was running checks. It was only until much later that the current scheduler saw the light of day. This scheduler was smarter in the sense that it did not run checks all at the same time, but organized them into a queue to keep the server running smoothly. At the time the difference was also more noticeable, since servers were slower and did not have multiple processor cores like nowadays. [Tuomo Karhapää, Chief Technical Officer, 14 April 2011, personal communication].

3.2 Nagios Objects

The various targets that the Nagios software monitors are called objects. They are defined in the configuration as services, service groups, hosts, host groups, contacts, contact groups, commands, time periods, notification escalations, notification dependencies and execution dependencies. [10, 4-5] Objects do not need to be defined as isolated entries. Object templates can be used when defining specific hosts, and the attributes defined in a template will be inherited by the object. Inheritance can also be used to split configuration to smaller pieces. For instance contacts and contact groups can be defined as their own entries, and then be used for a specific host or a service. [8, 6]

3.3 Nagios Checks

Nagios essentially builds on host checks and service checks. In the configuration one defines host objects and service objects. A host basically refers to a DNS name or an IP address (host address) and a service refers to an additional check made to the same host address. For instance, a host object could indicate an IP address 127.0.0.1 and a service object could be SSH. In this configuration setup Nagios would perform a host check on the local IP address and in addition would check that a TCP connection to port 22 was successful and that an SSH session could be established. An example of how this configuration looks like in the Nagios web interface is shown in Figure 3.

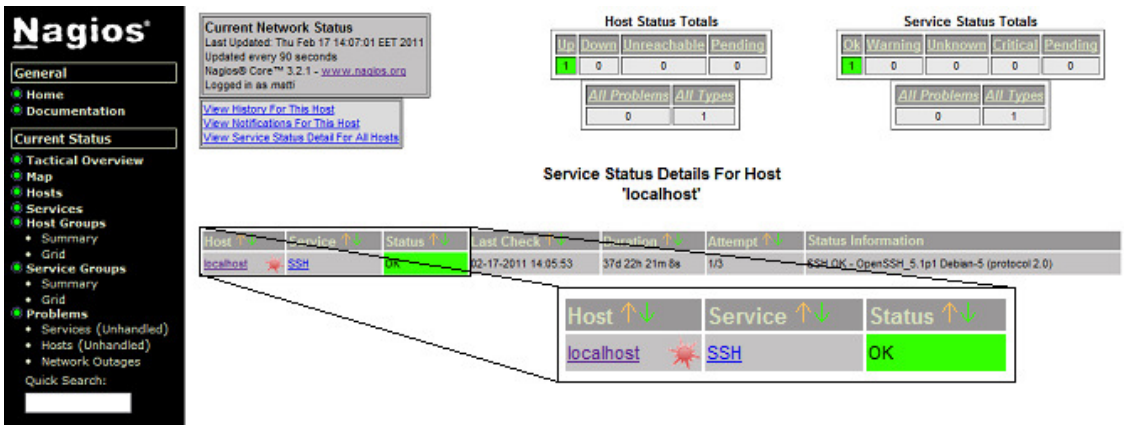

Figure 3: An example view of Nagios host and service check states.

The checks can again be split into two main categories: active checks and passive checks. An active check means that Nagios initiates the check in defined intervals to see what the result is. A passive check in turn means that Nagios itself does not initiate checks, but instead waits for a check result to be sent by an external process or application. [9]

3.4 Parent-Child Relationships

The parent-child relationships in Nagios are very important. A child host is dependent on another host considered as its parent host, meaning that if the parent host goes down, the child is also unreachable. In Nagios one only has to define parent hosts. Nagios then interprets these relations and forms a map from them. In addition to relationships between hosts, Nagios also handles host-service relationships. A host can exist with no services, but a service always has to be attached to a host. The parent-child relationship shows best in the sense that if a parent host generates an alert, Nagios sees the child hosts and services behind it and does not generate an alert for them. This is important if, for instance, a network switch breaks down. In this case Nagios only generates an alert for the switch, and does not flood the administrator's inbox with dozens of alerts about all hosts that become unavailable as a result. [10]

3.5 Nagios States

Nagios has four different states for service check results: OK, Warning, Critical and Unknown. These states in turn have a state type, which is either Soft or Hard. A soft

state means a service or host check has returned a value other than OK, but a notification has not been sent. The max_check_attempts configuration parameter controls how many times a check can return the same non-OK value before a notification is sent. For host checks the results are: Up, Down and Unreachable. Up and Down indicate the result of an actual host check, whereas Unreachable is a result determined from a host check of a parent host. If a parent host check fails, Nagios determines that its child hosts are also down, thus declaring them as Unreachable. [9]

3.6 Macros

Macros carry information that can be utilized in various parts of the configuration. For instance, when one defines a check command for ICMP ping, one can define the host address as a macro to the command configuration.

```
define host{
          host_name           localhost
          address             127.0.0.1
          check_command       check_ping
          …
          }

define command{
          command_name        check_ping
          command_line        $USER1$/check_ping -H
          $HOSTADDRESS$ -w 100.0,90% -c 200.0,60%
          }
```

Listing 2. Piece of Nagios configuration showing the use of macros.

In Listing 2 above, macros are used in the command definition to summon the file system path to Nagios plugin executables ($USER1$) and to provide a target argument to the command ($HOSTADDRESS$). In this case, the $HOSTADDRESS$ macro prints 127.0.0.1 as defined in the host definition for the address parameter. [9]

3.7 Nagios Configuration

Nagios configuration can be divided into four categories: main configuration, resources, object definitions and CGI configuration. The main configuration defines how the Nagios daemon operates. Resources consist of macros, that act as helper variables throughout the configuration. Object definitions include hosts, host groups, services, service groups, contacts, contact groups and check commands. CGI configuration defines how the frontend of Nagios operates. The configuration overview for Nagios is shown in Figure 4.

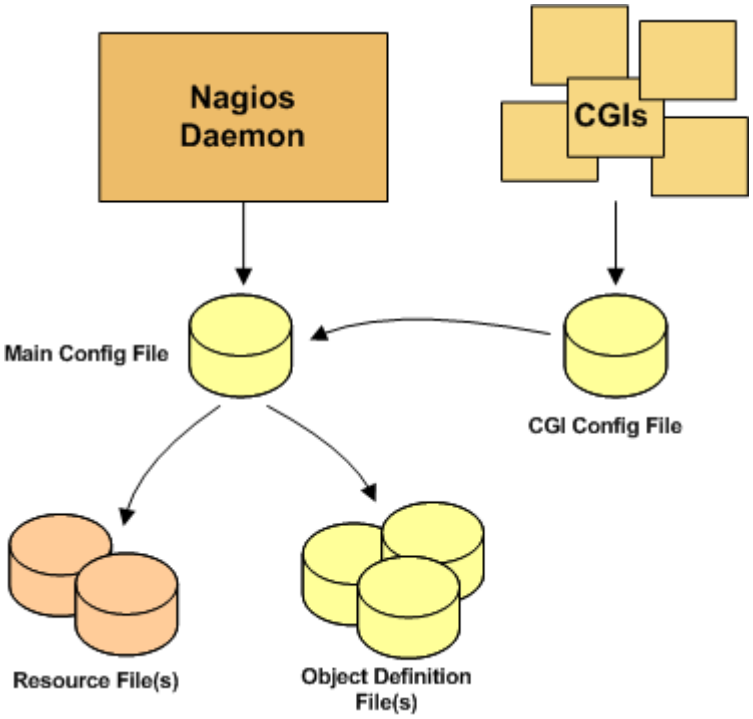

Figure 4: Nagios configuration overview. [9]

The Nagios source code package comes with example configuration files, where different parts of configuration are split into different files. These examples are very useful when creating one's first actual configuration, since they already define many different services based on different plugins. It is still best to figure out the best style of configuration for oneself, and not only replace the key values in these files. [9]

3.8 Nagios Web Interface

The Nagios web interface is where a user can view the current status for hosts and services. It provides a tactical overview screen that shows a summary of current problems, a map that shows the relations of hosts, host and service listings, host group and service group listings and problem listings. One can also run commands from the web interface. For instance, one can schedule downtime for a host or a service if one knows there is going to be a break, disable notifications for a host or a service that keeps going down for a known reason or acknowledge a problem that is being worked on. The Nagios web interface is shown in Figure 5. Reports can also be generated for a certain host or service and event history can be browsed to see the latest alerts.

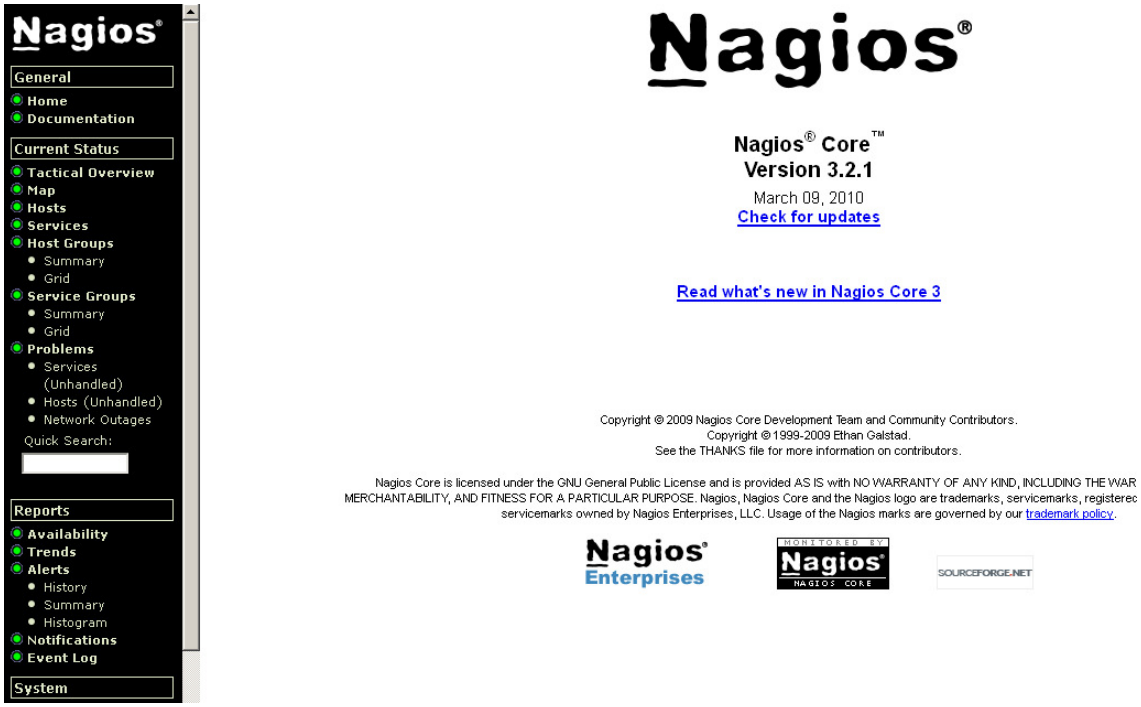

Figure 5: Nagios web interface.

Nagios does not include an HTTP server, but it utilizes other software, such as the Apache HTTP server. The Nagios web interface is built on Common Gateway Interface (CGI) executable files and static HTML files. CGI files need a special module in the Apache software to run. CGI files are basically small programs that produce HTML code. The CGI executable files read and interpret the Nagios status file and thus are

able to show current information about Nagios. [Tuomo Karhapää, Chief Technical Officer, 14 April 2011, personal communication].

3.9 PNP4Nagios

PNP4Nagios is an additional tool for Nagios that creates graphs from the performance data that Nagios plugins provide with check results. The data is actually stored in Round Robin Databases (RRDs) that can again be read to form graphs with different features. The PNP4Nagios web interface is not integrated to the Nagios web interface, but instead links are created to single host or service graphs in the Nagios view. The links to host and service graphs are shown in Figure 6.

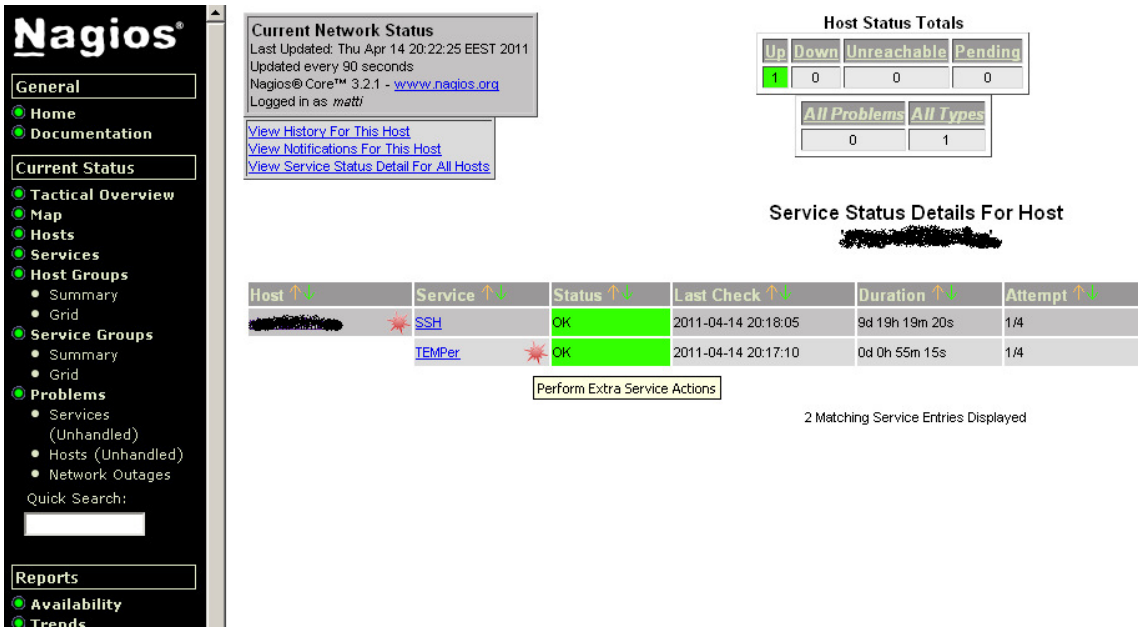

Figure 6: Links to PNP4Nagios graphs are shown as red icons next to the host or service.

The PNP4Nagios web interface shows the graphs in a defined time range. The number of graphs can also vary as defined in graph templates. Graph templates are PHP files where a user can define the outlook of the graphs. If a Nagios plugin returns many different values, the values can be drawn as separate lines or areas, or as their own graphs. PNP4Nagios graphs can also be exported as Portable Document Format (PDF) or Extensible Markup Language (XML) files. An example page of the PNP4Nagios interface is shown in Figure 7. [11]

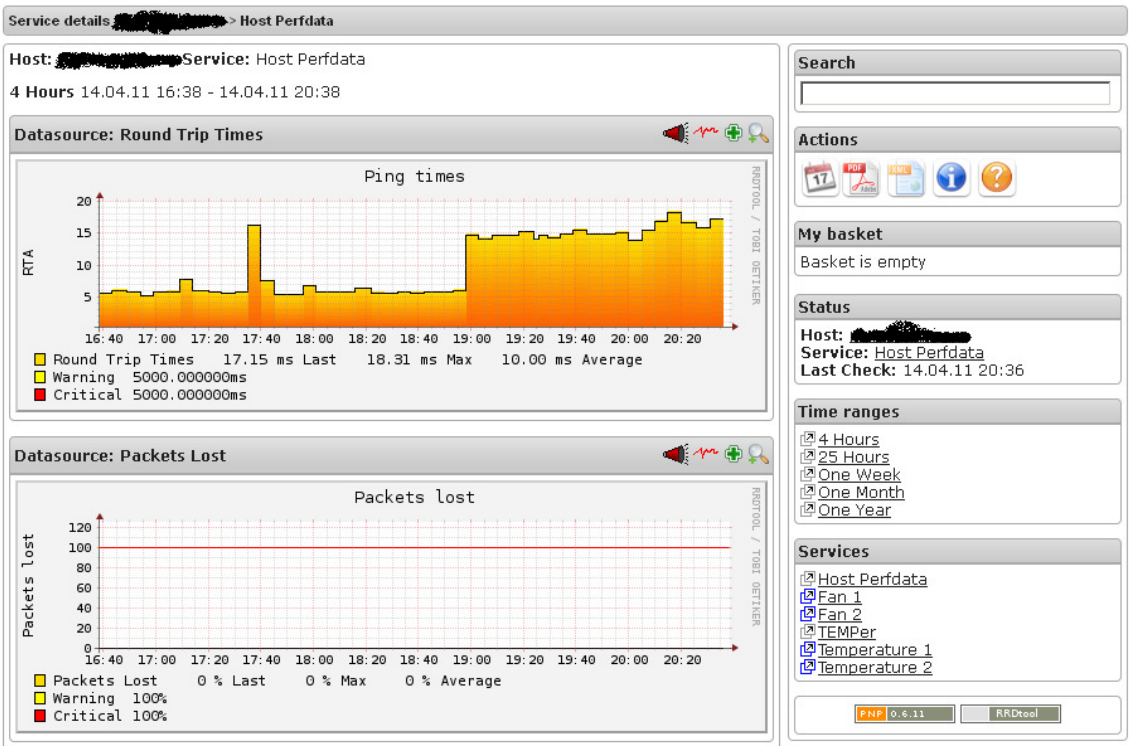

Figure 7: PNP4Nagios web interface.

The page shows data gathered by an ICMP ping plugin that is used as a host check command. On the top of the page the current time range is shown (4 Hours) and on the right side all available time ranges are listed. The links to PDF and XML export functions are also listed on the right side under the Actions title. [11]

4 Network monitoring at Otaverkko

4.1 Company

Otaverkko Oy is a growing Finnish company that provides Infrastructure as a Service (IaaS) and other IT services. A majority of the services are built on an infrastructure of three separate data centers and a high speed network that connects them to the company's headquarters and the Internet. The company is running and monitoring hundreds of physical and virtual servers, network switches and also several routers and firewalls.

In addition to data center services, the company provides Internet connectivity to a selection of other companies and organizations. Among these are Omnia – a vocational institute with some 10 000 students, and Laurea – a university of applied sciences with around 8 000 students. Otaverkko offers email and other Internet services to both insitutions. The company also provides monitoring for the primary name server of the Finnish top level domain (TLD) name service A.fi. [Ville Rindell, Production Manager, 8 March 2011, personal communication].

4.2 Nagios Server Platform

Otaverkko's Nagios servers run the Debian GNU/Linux operating system. It has been deemed a stable and sensible system and it is easy to control. The Nagios software – although available among the Debian package management system – has been installed separately from a source code package. This way it is easier to keep current with Nagios, and at the same time have a very stable operating system underneath. [Ville Rindell, Production Manager, 8 March 2011, personal communication].

4.3 Monitored Objects

4.3.1 Network Switches

Network switches used in a data center environment are almost always managed switches, meaning they can also be configured to have an IP address and generally are smarter than their unmanaged counterparts. Having a bit of intelligence is also vital for the detection of network loops and other problems. Otaverkko monitors the state of the switches with ICMP ping requests, and vendor specific values that are available, such as CPU utilization, memory usage and the physical condition of the device (fan speeds, temperatures and power supply states). In addition, network traffic graphing is also provided by a piece of software called Cacti. Figure 8 shows an example of a Cisco network switch.

Figure 8: Cisco Catalyst network switch. [14]

Since redundancy is important especially in a data center's core network, switches at Otaverkko are configured to provide a backup route for traffic in case the primary route becomes unusable. This means that switches form a connection loop, and must be configured properly to prevent traffic to create a short circuit. In the event of a network loop, traffic between switches fills up the bandwidth and packet loss increases. Thus a well configured ICMP ping check in Nagios creates an alert and notifies the network administrator. Otaverkko uses Extreme Networks and HP switches. Figure 9 shows the topography for a high availability network built with Extreme Networks switches. [Ville Rindell, Production Manager, 8 March 2011, personal communication].

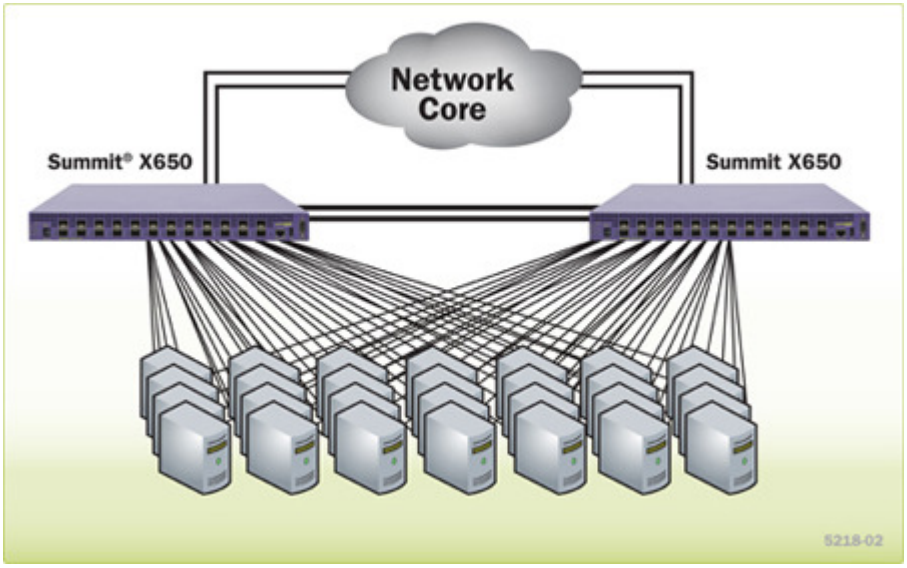

Figure 9: High availability network diagram. [15]

PNP4Nagios is used to store historical data from services such as ICMP ping, CPU usage and memory usage. Interface graphing is also implemented, but separately on a Cacti server. The Cacti software is used mainly for graphing, and not for monitoring and alerting. Listing 3 includes an example Nagios configuration for a switch at Otaverkko.

```
define host {
        use                     defaults
        host_name               sw1
        alias                   sw1
        address                 (deleted)
        hostgroups              ov-switches
        parents                 sw2
}

define service {
        use                     generic-ping
        host_name               sw1
}

define service {
        use                     extreme-env
        host_name               sw1
}
```

```
define service {
        use                     check_extreme_system
        host_name               sw1
}

define service {
        use                     check_uptime
        host_name               sw1
}

define service {
        use                     check_extreme_eaps
        host_name               sw1
}
```

Listing 3. Piece of Nagios configuration. The configuration shows how switches are monitored at Otaverkko.

This piece of configuration includes a host definition and several service definitions. To simplify the configuration these definitions make use of host and service templates, that set other parameters that can be shared among other devices' definitions. The use parameter in the above configuration defines the host or service that is used. Figure 10 shows what the Nagios web interface looks like for this configuration. Note that the figure also shows services that are not defined in the configuration above. These services are defined for the hostgroup to which this host entry belongs. [Ville Rindell, Production Manager, 8 March 2011, personal communication].

Host	Service	Status	Last Check	Duration	Attempt
[redacted]	EAPS Status	OK	04-11-2011 18:00:56	6d 1h 59m 28s	1/4
	Extreme PSU/Fan status	OK	04-11-2011 18:02:00	6d 13h 19m 9s	1/4
	Extreme System Status	OK	04-11-2011 18:02:33	236d 2h 54m 54s	1/4
	Interface graphing	OK	04-11-2011 18:03:11	261d 11h 3m 0s	1/4
	PING	OK	04-11-2011 18:02:44	6d 13h 18m 24s	1/4
	Uptime	OK	04-11-2011 18:04:32	144d 1h 9m 11s	1/1

Figure 10: Nagios configuration for a network switch.

4.3.2 Routers

Routers are at the very core of a data center network and direct traffic into the outside world and back. They require more processing power and intelligence than normal managed switches, because they handle routing protocols, perform VLAN translation and shape traffic so that the ISP connection stays usable. Otaverkko has two ISP connections for its data centers. If one edge router loses its BGP neighbor or becomes unreachable, the other edge router activates its BGP neighbor connection and starts routing traffic. Otaverkko monitors the routers' BGP states and notifies the network administrator if traffic is directed to the secondary ISP. This is important to find out as soon as it happens, as it may never be noticed otherwise. Not noticing this would create problems because the source of the problem still has to be discovered in case it happens again.

In addition to BGP routing, another routing protocol is usually used inside the ISPs network. In Otaverkko's case this is Open Shortest Path First (OSPF). The BGP routers advertise themselves to OSPF, so the internal routers know to direct traffic to the right direction in case of an ISP failover. Otaverkko also has a separate Funet connection for clients that are Funet members. Figure 11 shows the variety of Cisco Catalyst 6500 series routing network switches.

Figure 11: Cisco 6500 series routing switches. [14]

Otaverkko uses separate internal routing switches that handle switching between data centers, and internal routing. The internal routers have two management and switching modules (MSM) each, which in turn provide redundancy over each other. One module is always the master, and the other is the slave. If the master module for any reason becomes unavailable, the slave takes over its duties as the new master module. No suitable Nagios plugin for this was found online, so I had to create it myself. The following includes a piece of Nagios configuration that defines the use for this plugin. [Ville Rindell, Production Manager, 8 March 2011, personal communication].

```
define service {
        use                             defaults,srv-pnp
        name                            check_extreme_msm
        service_description             BlackDiamond MSM Status
        check_command                   check_extreme_msm!public
        process_perf_data               1
        register                        0
}
```

```
define command {
        command_name    check_extreme_msm
        command_line
/usr/local/nagios/libexec/check_extreme_msm.php -H
$HOSTADDRESS$ -C $ARG1$
}
```

Listing 4. Piece of Nagios configuration. The configuration shows the use an MSM redundancy plugin.

This quotation defines a service template, rather than an actual service object. The register configuration parameter makes sure that this service is not activated as a single object when set to 0. A definition for the plugin command is also defined below the service template. The command definition is called in the service template by the check_command parameter. The plugin in question here is used to check the MSM status. [Ville Rindell, Production Manager, 8 March 2011, personal communication].

4.3.3 Servers

Servers often require extensive monitoring that includes hardware health monitoring, performance monitoring and application-specific monitoring. Nowadays server manufacturers have developed sophisticated firmware that is installed on a chip that checks hardware status on fans, temperatures, hard disk states and much more. In order to read this information remotely, another piece of software needs to be installed on top of the operating system (OS) running on the server. This software again uses the SNMP agent (SNMP daemon) installed on the OS to share the hardware state along with queries or traps. Otaverkko uses server hardware made by some of the leading market manufacturers. An example of an HP BladeSystem c7000 is shown in Figure 12.

Figure 12: Hewlett-Packard BladeSystem c7000. [15]

Servers are often dedicated to one or two services. Depending on the services being run different kinds of methods can be used to monitor the condition of the services. For instance, in the case of a web server Otaverkko monitors TCP ports 80 and 443 (HTTP and HTTPS) and also usually a check is made that searches for a specific string on a website. This way an alert is sent if an underlying application server has crashed or if the website has been defaced. Another good implementation would be to monitor specific processes for their existence and used resources (CPU utilization and allocated memory). [Ville Rindell, Production Manager, 8 March 2011, personal communication].

4.3.4 Storage Appliances

Storage appliances are a critical part of a modern datacenter. They act as file servers sharing storage over IP networks but also provide faster and more reliable storage to servers over the Fibre Channel protocol. The monitoring for storage appliances is similar to monitoring of servers, but also concentrates on the performance of mass storage (hard drives and solid state drives). Another important aspect of monitoring

storage devices is following the level of used storage space for specific storage areas or network shares. Otaverkko uses NetApp and Hitachi storage systems. A Hitachi AMS2100 storage appliance is shown in Figure 13. [Ville Rindell, Production Manager, 8 March 2011, personal communication].

Figure 13: Hitachi AMS2100 without a bezel. [16]

Storage devices are critical in terms of monitoring as they provide services to many other devices, and also store valuable information. [Ville Rindell, Production Manager, 8 March 2011, personal communication].

4.3.5 Data Center Environment

Data center environment is monitored through temperature and humidity sensors. It is critical that the temperature stays low enough for servers to operate in optimal conditions. This ensures that the hardware keeps on running for a long time and without breakdowns. Another matter to take into account with monitoring a data center is the human factor or physical security. Cameras are placed in key points

throughout the room and alert administrators when motion is detected. An administrator can then watch the stored camera footage to determine whether a person is allowed to access the data center or not. [Ville Rindell, Production Manager, 8 March 2011, personal communication].

4.4 Alerting and Response

Otaverkko utilizes its technical staff to provide 24/7 monitoring and response to technical issues. Nagios sends out emails to everyone at the technical staff, and also text messages to whoever is on call at the time. That way everyone can react on their own behalf during office hours and the on-duty technician takes care of alerts outside office hours. Figure 14 shows an example of a Nagios e-mail alert message.

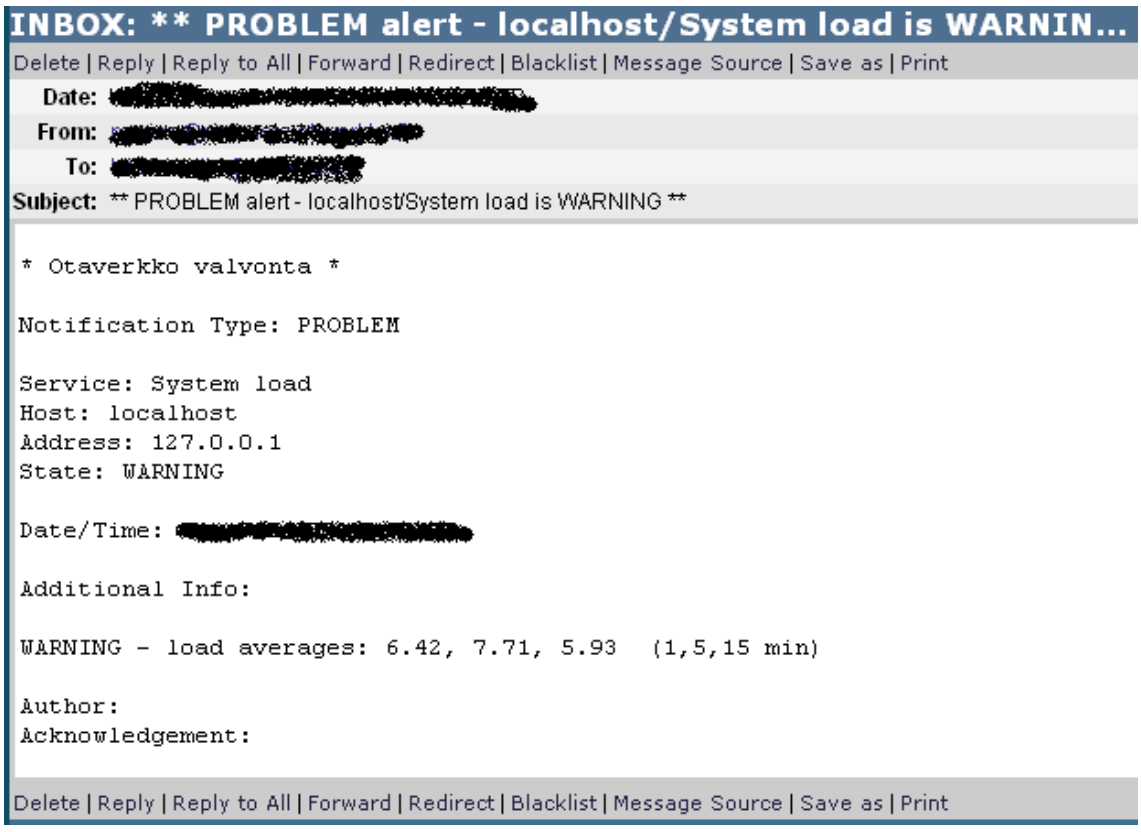

Figure 14: Nagios e-mail alert message.

The key components of the data center are monitored day and night and must be attended to within a few hours of a failure. Customer services and equipment are monitored depending on the contract. Some are more vital and must be monitored

constantly and others that are less important are monitored only during office hours, for instance.

Host checks (ICMP Ping) create an alert when a considerable amount of packets are lost during the operation or the round trip time is too high (several hundred milliseconds or more). Ports for different TCP services (HTTP, SMTP, SSH etc.) are usually either responding or completely dead, so an alert is generated in case a service is down. Hardware failures on servers are usually not as critical as other issues, since servers have redundant components such as mirrored hard drives, error correcting memory and multiple power supplies. [Ville Rindell, Production Manager, 8 March 2011, personal communication].

4.5 Creating new Plugins for Nagios

4.5.1 Doing the Research

As established earlier in sections 4.3.1 through 4.3.5, there are many different types of devices that need monitoring at a company such as Otaverkko. When I was told to tailor monitoring for a device that had not been monitored before, the first step I took was searching the Internet either directly for Nagios plugins for that particular device or for information about how to extract information from that device over SNMP or other means. A ready-made plugin was of course the easiest way, but many times I had to find the correct manufacturer-provided SNMP MIB file and browse it through for the information that was needed for efficient monitoring.

4.5.2 Inspecting a MIB File

An MIB file is easy to browse through provided that one has a good MIB browser available. If everything goes smoothly, all one needs to do is load the file into the browser and one can find the information needed in a tree-like format. Figure 15 shows an MIB browser for Unix systems that I used in my development efforts.

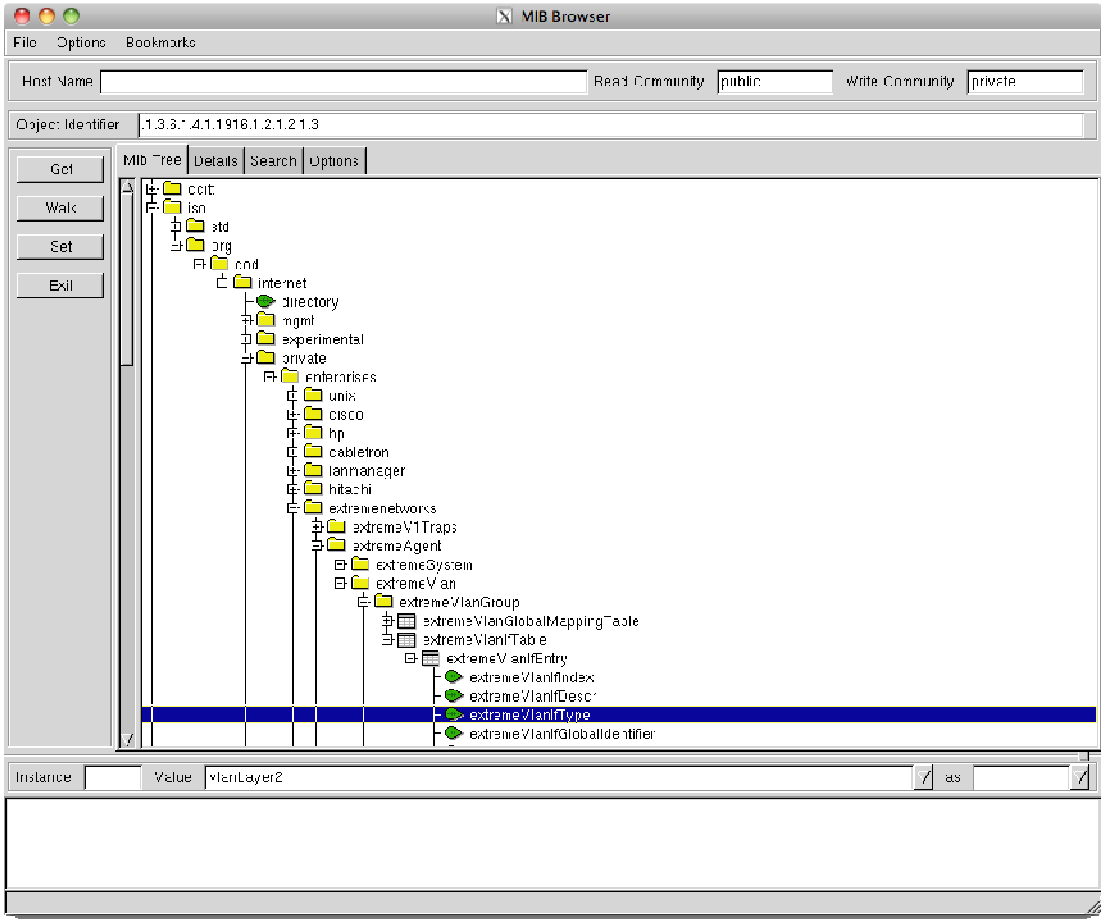

Figure 15: A Unix MIB browser.

Figure 15 above shows a tree structure with folders, tables and objects. These are marked respectively with a yellow folder icon, a white and grey table icon and a green object icon. An object named extremeVlanIfType is selected and above we can see its object identifier. The MIB browser in the figure is called mbrowse and is available as open source software.

4.5.3 Setting up and testing SNMP

After finding out the needed object identifiers I set up the device so that I could read SNMP objects' values with command line tools provided by the Net-SNMP software package. A particularly effective command in this package is called snmpwalk. With this command I could easily walk through a whole tree of information and compare the object identifiers and values to confirm that the MIB file was in fact correct. Listing 5 shows an example of the Linux shell command for snmpwalk.

```
snmpwalk -On -v2c -cpublic 127.0.0.1
1.3.6.1.4.1.1916.1.2.1.2.1.2
```

Listing 5. Example use of the snmpwalk Linux shell command.

The first configuration switch (-On) defines which form to use for object identifier and value output, in this case n as in numeric. The second switch (-v2c) defines the version of the SNMP protocol to use, which is version 2c and the third switch defines the SNMP community string to use, which is public. Finally the IP address 127.0.0.1 is shown followed by an object identifier in its numeric form.

4.5.4 Creating the Code

When I had tested that I could read the objects I needed to use, I moved on to the actual programming part of the plugin development process. Having had more experience with PHP than with Perl (the native programming language in Nagios plugin development) I felt comfortable creating my own code from scratch, and did not try to find a suitable framework to use. Granted, having spent some time learning the differences between the two languages, I could have been more efficient in the long term as a Nagios plugin developer. However since the need for plugins comes and goes, it felt easier and faster to just do everything tailored as separate PHP scripts.

5 Conclusions

5.1 Results and Current State

The monitoring system at Otaverkko was already in good form when I was hired for my job. Many different devices and services were already being monitored, so it was up to me to analyze and further develop the system that was already in production. However, the system needed to be refurbished since much of the work was done a few years ago. What I did solidified the system and I also discovered some features that were more or less important to monitor. Thanks to the skills I learned in my previous job as a software integrator, I was also able to program new plugins where no ready or suitable plugins were found.

Finding out what the configuration already did took some time, because of the large number of Nagios configuration files and the amount of data in them. While I was looking through the files, I noticed command and service definitions that had been added on top of the other making the other one useless. I also cleaned out some pieces of configuration that were still lying around because they had not been cleaned right after their obsolescence.

When I started my job, there already were two independent Nagios installations. However, the monitoring of the Nagios process was not implemented. I discovered that there was a plugin available to monitor the Nagios status file that includes all of the data about monitored objects in Nagios, and is updated frequently. The plugin created an alert if the file modification timestamp was older than the configured update interval. This was a very important improvement to the monitoring setup, since the Nagios processes do crash sometimes (although very seldom).

Much of the monitoring was done using plugins that were already available through the Internet, but I discovered that there was much other information available when one queried the devices with SNMP. This information included environmental readings on certain devices, the number of open connections in a firewall appliance and general status queries on a device's operational state, among many others.

5.2 Suggestions for Improvement

The work is still ongoing, and will remain such because the industry itself is growing and changing as time passes. Nagios is being developed further with new features that may prove useful and new devices will be deployed throughout the network as new standards emerge. As this is not my only duty at the company, there is still work to be done with the current implementation and network structure. For instance, the web views for multiple Nagios instances could be joined, more Nagios instances could be deployed among different virtual LANs to share the load, and Windows servers could be monitored better with Windows Management Instrumentation (WMI).

Nagios of course is not the only choice on the market, and it has also its own descendants: Icinga and Centreon. Other rivals include Zenoss, Zabbix and OpenNMS. Among these alternatives, Nagios still seems like the most stable and flexible system in my eyes. This is due to the simple nature of text-form configuration files, the endless supply of plugins available on the Internet and its use of the ancient yet most widely used C and Perl programming languages. In addition to its merits, Nagios also has a wide user base which provides a solid ground for troubleshooting.

References

1 Simple Network Management Protocol [online]. Wikipedia. URL: http://en.wikipedia.org/wiki/Snmp. Accessed 17 February 2011.

2 Tang, R. How to create SNMP Test Trap on Linux [online]. Ryan's Tech Notes. URL: http://technotes.twosmallcoins.com/?p=369. Read 15 April 2011.

3 Management Information Base [online]. Wikipedia. URL: http://en.wikipedia.org/wiki/Management_Information_Base. Accessed 17 February 2011.

4 Cisco Systems. SNMPv3 [online]. Cisco Systems official website. URL: http://www.cisco.com/en/US/docs/ios/12_0t/12_0t3/feature/guide/Snmp3.html. Accessed 12 April 2011.

5 SNMP Research International Inc. Security in SNMPv3 versus SNMPv1 or v2c [online]. Louvain-le-Neuve, Belgium: SNMP Research International Inc.; 2002. URL: http://www.aethis.com/solutions/snmp_research/snmpv3_vs_wp.pdf. Accessed 7 April 2011.

6 Network management station [online]. Wikipedia. URL: http://en.wikipedia.org/wiki/Network_management_station. Accessed 15 April 2011.

7 Nagios History [online]. Nagios official website. URL: http://www.nagios.org/about/history. Accessed 17 February 2011.

8 Schubert, M., Bennett, D., Gines, J., Hay, A., & Strand, J. Nagios 3 Enterprise Network Monitoring Including Plug-Ins and Hardware Devices. Burlington: Syngress Publishing, Inc., 2008.

9 Nagios Core Version 3.x Documentation [online]. Nagios official website. URL: http://nagios.sourceforge.net/docs/nagioscore/3/en/. Accessed 17 February 2011.

10 Nagios: Establishing parent – child relationship is both easy and required [online]. Nagios official website. URL: http://community.nagios.org/2009/08/23/nagios-establishing-parent-child-relationship-is-both-easy-and-required/. Accessed 14 April 2011.

11 Documentation [online]. PNP4Nagios website. URL: http://docs.pnp4nagios.org/pnp-0.6/start. Accessed 14 April 2011.

12 Cisco network switch price list [online]. India Price List official website. URL: http://www.indiapricelist.com/?p=807. Accessed 13 April 2011.

13 Summit X650 series [online]. Extreme Network official website. URL: http://www.extremenetworks.com/products/summit-x650.aspx. Accessed 15 April 2011.

14 Cisco Catalyst 6500 Series Switches [online]. Cisco Systems official website. URL: http://www.cisco.com/en/US/products/hw/switches/ps708/prod_view_selector.html. Accessed 14 April 2011.

15 HP BladeSystem c7000 [online]. Flickr. http://www.flickr.com/photos/h_u_p/2636339474/. Accessed 15 April 2011.

16 Gateway AMS2100 [online]. Gateway Netherlands official site. URL: http://nl.gateway.com/products/product.html?prod=AMS2100. Accessed 15 April 2011.

17 Mbrowse [online]. Mbrowse SourceForge project website. URL: http://mbrowse.sourceforge.net/. Accessed 15 April 2011.